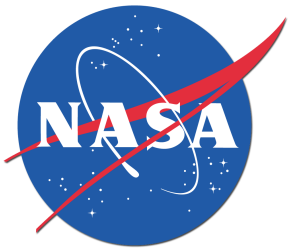

User Guide

NASA Unified WRF (NU-WRF)

Release 3-3.2.1

Document Revision: 1
November 10, 2011

Table of Contents

Overview

The Weather Research and Forecasting (WRF) model is a next-generation mesoscale weather forecast and assimilation system designed for portability, efficiency, and applications in both research and operations. WRF can be used for a wide range of configurations ranging from idealized large-eddy simulations to global modeling, with an emphasis on horizontal grid sizes in the range of 1–10 km. For large domain applications such as regional weather and climate processes, WRF is typically run within a domain covering several thousand kilometers using interactive nesting techniques with multiple grid refinements. See http://www.mmm.ucar.edu/wrf/users/docs/arw_v3.pdf for technical details on the official WRF system.

At NASA, WRF is used to study a variety of phenomenon such as high impact weather and aerosol effects. Several parameterizations of physical processes developed by NASA scientists have been implemented in WRF to better represent/simulate cloud-aerosol-precipitation-land surface processes. NASA Unified WRF (NU-WRF) combines these improvements in order to: (1) facilitate better use of WRF in scientific research; (2) reduce redundancy in major WRF development at NASA; (3) prolong the serviceable life span of WRF; and (4) allow better use of NASA high-resolution satellite data for research (short term climate and weather).

Enhancements

NU-WRF version 3-3.2.1 was released as beta software in November 2011. The version number "3-3.2.1" (or more simply, "release 3") designates the third official release of NU-WRF, and is based upon the WRF 3.2.1 code base. The NU-WRF software project consists of multiple software packages and enhancements built around the WRF modeling system, use cases with input data sets, and documentation describing the software and software processes.

Current Components

NU-WRF software consists of the following software components (new features highlighted in blue):

- WRF 3.2.1 (including WRF/Chem 3.2.1) with NASA enhancements:
    - Updated Goddard microphysics
    - Updated Goddard radiation (longwave and shortwave)
    - On-line coupling between GOCART aerosols and Goddard radiation
    - On-line coupling between GOCART aerosols and Goddard microphysics
    - On-line coupling with NASA Land Information System (LIS) 6.1
    - Estimation of SOA from biogenic terpene emissions
    - Linking of MEGAN2 biogenic emissions scheme to GOCART related chemistry
    - Linking of RADM2 chemistry to GOCARTRADM2 option;
    - Linking various optical property schemes to GOCART related chemistry
    - Linking GOCART dry deposition scheme to GOCART related chemistry
    - “Sanity check” module to identify unphysical values passed between coupled

physics parameterizations
- Severe weather diagnostics (lightning flash rate, etc.)

- WPS 3.2.1 (Standard WRF pre-processors for terrain, land-use, and initial/lateral boundary conditions) with enhancements:
    - Fixed processing ERA-Interim sea surface temperatures
- WPP 3.2 (standard WRF post-processor/GRIB generator) with additions:
    - Severe weather diagnostics
- MET 3.0 (verification package)
- ARWpost 2.2 (WRF post-processor/GrADS and Vis5D file generator)
- RIP 4.6 (WRF post-processor/NCAR Graphics plotting software)
- NASA LVT (LIS Verification Toolkit)
- Goddard SDSU 2.0 (simulates satellite measurements from WRF gridded data)
- prep_chem_sources (preprocesses fire emissions) with additions:
    - Supports GFEDV3 fire emissions
    - Added Mercator map projection
- NASA data conversion utilities:
    - GEOS2WRF (for NASA GEOS-5 global model output)
    - MERRA2WRF (for NASA MERRA reanalysis output)
    - GOCART2WRF (for NASA GOCART global aerosol model output) with bug fixes and modifications to support GEOS-5 GOCART data
    - SST2WRF (for sea surface temperature analyses from Remote Sensing Systems)
    - LISConfig (customizes LIS domain to match that of WRF)
    - fromGEOS5_to_GEOS4 (temporary software package to convert GEOS5 GOCART netCDF4 aerosol data to GEOS4 GOCART netCDF3 format)
- Unified build system for compiling software components (excluding temporary fromGEOS5_to_GEOS4 software):
    - Supports building NU-WRF software on both Discover and Pleiades
    - Added build support for conv_emiss preprocessor bundled with WRF/Chem (although the preprocessor is not complete)

Vendor-Specific Updates

No vendor updates have been incorporated in Release 3 over those in Release 2.

NU-WRF Group Improvements

This release of NU-WRF adds a number of improvements to both the third-party software packages and in-house enhancements.

Land Information System (LIS)

- LIS updated to v6.1 rp1 updates include:
    - Made landcover consistent with UMD and USGS, added two diagnostics T2 and Q2, updated to use ESMF 3.1.0rp3 library and NU-WRF MPI library, other minor changes and fixes related to greenness fraction, offline use of background and actual roughness length, and forcing height.
    - Support for SPoRT Daily GVF data, North American Regional Reanalysis (3d)

(NARR) data, NCEP's modified IGBP MODIS landcover data, and specify direction for output variables. Fixes a problem affecting non-square lat/lon grids.
    - Changes to make Noah 2.7.1, Noah 3.1, and Noah 3.2 consistent and add fixes for handling of exchange coefficients, running Noah 3.2 in LIS-WRF with nesting enabled, catchment, running nests in non-ESMF coupled mode, writing NetCDF when running in parallel, overwriting CH and CM coefficients.
- Added METGRID.TBL file for use with LIS soil data and LIS Vtable (assumes Noah LSM is run).
- Adds LIS Verification Toolkit (LVT) added into the standard NU-WRF packages. Builds on Discover as part of the NU-WRF automated build system.
- Adds LIS-to-WRF configuration utility is added to support automatic generation of lis.config files that match the WRF/WPS grid.

Goddard Radiation

Goddard radiation and GOCART are coupled by default when running with WRF/Chem. The radiative transfer model now factors in aerosols, such as elemental carbon, sulfate, sea salt, and dust, in the short and long-wave radiation. A new switch ('gsfcrad_gocart_coupling') has been added to the 'chem' namelist to disable coupling between aerosols and radiation.

Additional fixes include:

- Prevents negative effective radii and division-by-zero.
- Adds special handling of spurious negative mixing ratios.
- Includes sanity checking routines mostly for debugging purposes, but can be expanded to support other areas of the code.
- Bug fix to the Grell-Devenyi cumulus scheme.

Goddard Microphysics

The Goddard microphysics has been updated and is now coupled by default with GOCART aerosols when GOCART chemistry is activated. The microphysics update includes addition of CCN and IN -- either from look-up tables or from GOCART aerosols -- and discrimination between convective and stratiform precipitation. A new switch ('gsfcgce_gocart_coupling') has been added to the 'chem' namelist to disable coupling between aerosols and microphysics. It includes a new ice microphysics scheme as well.

WRF/Chem/GOCART

- Added capability of estimating SOA from biogenic terpene emissions, including three new variables: e_terp, e_api, e_lim
- Linked the following components to GOCART chemistry: MEGAN2 biogenic emissions scheme, GOCART dry deposition scheme, and various optical property schemes
- Linked RADM2 chemistry to GOCARTRADM2 option
- Fixed bugs in various emissions modules

Goddard SDSU

- Renamed folder to "GSDSU" to differentiate from HyARC version.

- Adds GOCART input options to WRF input, GCE-SBM 3D option, SBM moment output, grads control file automatically written, and Morrison two-moment support.
- Also adds bug fixes.

Data Utilities

- Added "fromGEOS5_to_GEOS4" utility to support on-line GEOS-5 GOCART data to off-line GEOS-4 GOCART format for gocart2wrf, including conversions of several mixing ratios to mol/mol. This is a temporary utility with some hard-wired settings. A future version of gocart2wrf should support both data natively.
- Added "prep_chem_sources" utility based on a third-party source code for use of GFEDv3 fire dataset. Support for the conv_emiss program has been added. This program was already bundled with the WRF/Chem software, and allows users to read in emissions data from prep_chem_sources and vertically interpolate to the WRF grid.Handles new and old GOCART background files (set by namelist variable "gocart_bg_data_type". The Mercator map projection has also been added. With these changes, it is now possible to horizontally interpolate emissions (fire, biogenic, volcanic, and aerosol) data to the WRF grid.
- Modified convert_emiss to be compatible with WRF3.3 version of prep_chem_sources by adding read sequence for NO2 after SO2 and ASH after SULF.
- A new "LISConfig" Fortran program and accompanying Python script have been written to aid configuring LIS as input to WRF. Currently the software takes a netCDF file generated by the WPS preprocessor METGRID, extracts the map projection and grid information, and updates a lis.config file so that the LIS domain matches the WRF domain.
- Added "fromGEOS5_to_GEOS4" utility to support conversion of GEOS5 GOCART netCDF4 files into GEOS4 GOCART netCDF3 look-alikes, including conversions of several mixing ratios to mol/mol. This is a temporary utility with some hard-wired path settings in the source code. A future version of GOCART2WRF should support both data natively.
- GOCART2WRF updated. A new option was added to correctly process files from fromGEOS5_to_GEOS4 (use different pressure calculation and vertically flip the data). Bug fixes were implemented to correctly process dust species and to correctly handle when a WRF and GOCART level coincide. Fixed bug triggered when WRF and GOCART pressures are identical and added minor optimizations. Also Supports converted GEOS-5 data (e.g. GEOS-5 data are vertically flipped.) Now works with inner-nest domains, but each must be processed separately. Also writes pressures of GOCART levels in each WRF column (to file p_gocart_3d) for use in convert_emiss, which is currently forced to uniform 1023 mb. (*Note: A next-generation GOCART2WRF is under development to better address these issues and to improve performance, but this new program is not finished yet.*)

Build System

The unified build system continues to support both the NCCS Discover supercomputer (at NASA Goddard Space Flight Center) and now adds support for the NAS Pleiades supercomputer (at NASA Ames Research Center). The implementation has changed to

automatically detect the local computer environment and select a default configuration file for that environment. The user can override by using the --config flag to specify a different file. The configuration files have been updated to use ESMF 3.1.0rp3 and other external libraries. New build targets include "conv_emiss", "prep_chem_sources", "lisconfig" and "lvt".

Additional changes include:

- Build support for incremental building as the default option (instead of a performing a complete rebuild. Note that in certain cases a full rebuild of WRF must still occur. Build also adds these new targets:
    - "allclean" - deletes all executables and intermediates
    - "clean" - deletes specified executables and intermediates
- Updated Discover configuration to use Intel MPI 4.0.1.007-beta.
- Rebuilt and configured all NU-WRF libraries for SLES 11 on Discover and also did similar upgrade on Pleiades.
- Added build script that helps install/build external NU-WRF libraries.
- Patched configure scripts for NCAR Graphics 6.0.0 and BUFRLIB 10.0.1.
- Build convert_emiss when 'chem' target is selected.
- Eliminated some false compiler errors and made other minor fixes.

Configuration

- Added chemistry namelist options to support optional couplings:
    - gsfcgce_gocart_coupling: GOCART aerosols and GSFC microphysics
    - gsfcrad_gocart_coupling: GOCART aerosols and GSFC radiation
    - bio_emiss_soa: emission conversion to SOA (off by default)
- Adds NU-WRF configuration templates for WRF and LIS under: WRFV3/test/em_real and WRFV3/lis/testcases/nuwrf
- WRF registry update significantly reduces overall chemistry outputs

Note that this documentation is also available from the "CHANGELOG.TXT" file in the NU-WRF source.

Using the Software

Acquiring the Source Code

Reference the *NU-WRF Source Code and Data* documentation at:
https://modelingguru.nasa.gov/docs/DOC-1834

Building the Source Code

The WRF modeling system is composed of a number of software packages, each of them with their own separate build system. To make it easier for the user to create desired executables and to more easily resolve dependencies between packages, the NU-WRF Release 3 includes a set of high-level "wrapper" scripts for building. With this new system, the user can build executables using a single script called **`build.sh`** located in the top-level directory. This script

accepts three types of command-line arguments:

- *Configuration*. The **`--config`** flag followed by the name of the configuration file specifying system specific environment variables (e.g., the path to the netCDF library). Currently two files are included in the top-level directory of the distribution: **`discover.cfg`** and **`pleiades.cfg`**. Users may develop their own configuration file to customize their settings. If no choice is given for the configuration file, the script will default to either **`discover.cfg`** or **`pleiades.cfg`** based on the local environment.
- *Options.* The user may specify **`lisreal`** and/or **`cleanfirst`**. The **`lisreal`** option indicates that if WRF is built, then the resulting `real.exe` executable will generate initial conditions for LIS. The **`cleanfirst`** option will cause the build system to "clean" a target (delete object files and static libraries) before building it.
- *Full Build Targets*. The user can specify any of these to build the entire system:
  - **`all`** (Build all executables w/o chemistry)
  - **`allchem`** (Build all executables w/ chemistry)
- *Clean Targets.* The user can use these to remove build products.
  - **`clean`** (Delete specific target's executables and intermediates)
  - **`allclean`** (Delete all executables, object files, and static libraries)
- *Specific Component Targets*. The user can specify one or more of these targets for the build system during a build or regular clean step:
  - **`wrf`** (Builds executables in WRFV3 directory w/o chemistry, excluding conv_emiss)
  - **`chem`** (Builds executables in WRFV3 directory w/ chemistry, including conv_emiss)
  - **`wps`** (Builds executables in WPS directory)
  - **`wpp`** (Builds executables in WPPV3 directory)
  - **`rip`** (Builds executables in RIP4 directory)
  - **`arwpost`** (Builds executables in ARWpost directory)
  - **`geos2wrf`** (Builds executables in utils/geos2wrf directory)
  - **`merra2wrf`** (Same as geos2wrf target)
  - **`gocart2wrf`** (Builds executables in utils/gocart2wrf directory)
  - **`sst2wrf`** (Builds executables in utils/sst2wrf directory)
  - **`prep_chem_sources`** (Builds executables in utils/prep_chem_sources directory)
  - **`met`** (Builds executables in MET directory)
  - **`sdsu`** (Builds executables in sdsu directory)
  - **`lvt`** (Builds executables in LVT directory)
  - **`lisconfig`** (Builds executables in utils/lisconfig directory)
- *Clean Targets*. The user must specify one or more targets for the build system:

In practice, the **`build.sh`** script may (re)build additional targets to resolve dependencies between programs.

Basic Build on Discover or Pleiades

The most straight-forward way to build the full system on Discover or Pleiades is to run the build script from the top-level folder:

```
./build.sh all
```

Or to build with chemistry enabled:

```
./build.sh allchem
```

To build a special version of `real.exe` to produce LIS initial conditions, the **`lisreal`** option must be included:

```
./build.sh lisreal all
```

And to fully clean the package, type:

```
./build.sh allclean
```

Users can also explicitly specify the configuration file, e.g.,

```
./build.sh --config discover.cfg lisreal all
```

Note that the build system will automatically select the **`discover.cfg`** (**`pleiades.cfg`**) if it detects the software is being built on Discover (Pleiades). The **`--config`** flag option is intended for future development and porting.

Selectively Building Targets

The build script will run a build on each target of NU-WRF. Because NU-WRF consists of multiple packages, this may take a while to finish. The user can selectively build packages by listing specific targets. For example, to build the WRF model on PLEIADES without chemistry along with WPS and WPP:

```
./build.sh wrf wps wpp
```

About the Build System

The top-level **`build.sh`** calls lower-level `build.sh` wrapper scripts located in each package directory (WRFV3, WPS, etc.). Configuration settings are passed to the lower-level scripts via environment variables. Each lower-level script is customized to directly manage the component-specific build mechanism (e.g., the `configure` and `compile` scripts for WRFV3, `make` for utils/geos2wrf), and to inject the appropriate configuration settings into that build mechanism. For example, the `build.sh` for WPS will modify the `configure.wps` file generated by `configure` to update several library paths; the modified `configure.wps` is then used by the `compile` script.

One complication addressed by the build system is that several packages (WPS, WPPV3, and ARWpost) are dependent on libraries and object files from WRFV3. In addition,

WPPV3 requires WRFV3 to be built in serial mode without chemistry. To account for these dependencies, the build script has the following behavior:

- If WPPV3 needs to be built, WRFV3 will be checked first to see if it has already been built in serial mode without chemistry. If it hasn't, WRFV3 will be cleaned and built with these settings, and then WPPV3 will be built. **This procedure will always occur before any other executables are built. This will occur even if the `wrf` target is not listed on the command line.**
- If WPS or ARWpost need to be built, WRFV3 will be checked to see if it was built in MPI mode. The required WRFV3 object files and static libraries will also be checked to see if they exit. If conflicts are found in the previous build and/or files are missing, WRFV3 will be cleaned and then automatically built in MPI mode before WPS and/or ARWpost. **This will occur even if the `wrf` target is not listed on the command line.**

An additional complication is specific to NU-WRF: The coupling of LIS to WRF introduces several new library dependencies on WPPV3, including ESMF 3.1.0rp3. Furthermore, since WPPV3 expects WRFV3 to be compiled serially, the ESMF library must likewise be built in serial mode. To account for this, two separate ESMF libraries are specified in the configuration file, one for each compilation mode. Also, the template configuration for WPPV3 is modified to add LIS-specific libraries so all necessary routines are resolved.

Currently the build configuration files for both Discover and Pleiades are set to use Intel compilers and Intel MPI. While support for other compilers and/or MPI implementations will be considered, implementing such support will probably require low-level changes to the build system.

Running NU-WRF

For a general guide to the WRF software, reference the *WRF on Discover* documentation at: https://modelingguru.nasa.gov/docs/DOC-1671

For information on NU-WRF use cases, reference the *Running NU-WRF Use Cases* documentation at:
https://modelingguru.nasa.gov/docs/DOC-1883

Standard Configurations

The NU-WRF team has collectively decided on a base set of settings that capture critical variables, leverage the capabilities of the NU-WRF software and allow for improved comparisons between runs and experiments. These configurations are located in the following directories:

- **WRFV3/test/em_real/namelist.input.nuwrf.diurnal** - standard configuration for non-chemistry runs that balances day/night concerns
- **WRFV3/test/em_real/namelist.input.nuwrf.chem** - standard configuration for WRF/Chem runs. This configuration includes NU-WRF-specific variables for SOA

bioemissions, Goddard microphysics-GOCART and Goddard radiation-GOCART. Note that the bioemissions is off by default as it requires additional data.

Additionally, LIS has its own independent configurations that can be used:

- **WRFV3/lis/testcases/nuwrf/lis.config.CONUS1_1KM.nuwrf** - NESDIS greeness data
- **WRFV3/lis/testcases/nuwrf/lis.config.CONUS2_1KM.nuwrf** - SPORT greeness data
- **WRFV3/lis/testcases/nuwrf/lis.config.GLOBAL_1DEG.nuwrf** - outside the CONUS

Known Issues

The following issues are known in NU-WRF:

1. *CFL errors.* With the introduction of the WRF 3.2.1 code base at least one of the supported use cases (Case 9) produces CFL errors with the Intel compiler 10 and higher. This occurs with the official WRF 3.2.1 as well. This problem has been reported to NCAR and Intel. No known workaround exists (except perhaps switching to the PGI compiler, which is not supported in NU-WRF).
2. *Namelist changes.* In particular the "grid_fdda" setting has a new behavior introduced from WRF 3.2.1 which can cause errors. To turn off gridded fdda for a given domain, namelist variables grid_fdda, gfdda_end_h, and gfdda_interval_m must all be set to zero.
3. *Vertical interpolation of moisture*. The `real.exe` program (part of WRF) does not correctly process specific humidity (QV) from `metgrid.exe` (part of WPS). Moisture must be converted to relative humidity before processing by `real.exe`.
4. *First output dump missing when restarting WRF*. When using restart files, WRF will not produce the initial wrfout file. This behavior is also found in the official WRF. There is no known workaround.